CLAIMS

1. A method of adding a computer file to a database, said method comprising:

receiving said computer file to be added to said database;

computing a first hash value for said file using a first hash function;

computing a second hash value for said file using a second hash function;

storing said file in said database at a location identified by said first hash value; and

adding said first hash value and said second hash value to a data structure associated with said database, wherein said first and second hash values are associated with one another in said data structure, whereby said addition of said first and second hash values to said data structure indicates the presence of said file in said database.

2. A method as recited in claim 1 further comprising:

determining whether a copy of said file is present in said database;

when it is determined that a copy of said file is present in said database, computing a verification hash value for said copy of said file using said second hash function and comparing said second hash value to said verification hash value; and

taking an action when it is determined that said second hash value matches said verification hash value.

3. A method as recited in claim 1 further comprising:

determining whether a copy of said file already exists in said database by searching said data structure for a value that matches said first hash value.

4. A method as recited in claim 2 further comprising:

determining whether a copy of said file is present in said database by searching said data structure for a value that matches said first hash value.

5. A method as recited in claim 1 wherein said data structure is a table, a list or a tree data structure.

6. A method as recited in claim 1 wherein said second hash function provides stronger security than said first hash function.

7. A method of retrieving a desired computer file from a database, said method comprising:

obtaining a unique identifier for said desired computer file;

retrieving a stored file from said database using said unique identifier as a reference;

retrieving a first hash value-second hash value pair from a data structure associated with said database by using said unique identifier, said unique identifier matching said first hash value, wherein said first hash value has been derived from said stored file using a first hash function and wherein said second hash value has been derived from said stored file using a second hash function;

computing a verification hash value for said stored file using said second hash function;

comparing said verification hash value to said second hash value; and

determining that said stored file is said desired file when said verification hash value matches said second hash value, whereby said desired file is retrieved from said database.

8. A method as recited in claim 7 further comprising:

computing an addressing hash value for said stored file using said first hash function;

comparing said addressing hash value to said first hash value; and

indicating that said second hash value matches said addressing hash value.

9. A method as recited in claim 7 further comprising:

when it is determined that said stored file is said desired file, delivering said stored file to a user.

10. A method as recited in claim 7 further comprising:

performing said step of retrieving a stored file by searching said data structure for said unique identifier.

11. A method as recited in claim 7 wherein said data structure is a table, a list or a tree data structure.

12. A method as recited in claim 7 wherein said second hash function provides stronger security than said first hash function.

13. A method of adding hash authority functionality to a database of files, said method comprising:

creating a hash authority data structure, said data structure including a plurality of entries;

retrieving an addressing hash value for a first file of said database, said addressing hash value having been computed from said first file using an addressing hash function;

retrieving said first file from said database;

computing a verification hash value for said first file using a verification hash function;

adding said addressing hash value and said verification hash value to one of said entries of said hash authority data structure, said addressing hash value and said verification hash value being associated with one another in said database.

14. A method as recited in claim 13 wherein said hash authority data structure is a table, a list or a tree data structure.

15. A method as recited in claim 13 wherein said verification hash function provides stronger security than said addressing hash function.

16. A method as recited in claim 13 further comprising:

performing the final four steps of claim 13 for each of the other of said plurality of files in said database.

17. A method of upgrading a verification hash function in a database of files, said method comprising:

accessing a data structure representing said database of files, said data structure including an addressing hash value-verification hash value pair for each of said files in said database, wherein said addressing hash values have been computed using an addressing hash function, and wherein said verification hash values have been computed using said verification hash function; and

for each of said files in said database,

retrieving the file,

computing a new verification hash value for the file using a stronger verification hash function, and

adding said new verification hash value to said data structure in association with said addressing hash value of said file.

18. A method as recited in claim 17 wherein said data structure is a table, a list or a tree data structure.

19. A method of adding a computer file to a database using a random number generator, said method comprising:

receiving said computer file to be added to said database;

generating a random number for said file using said random number generator;

computing a verification hash value for said file using a verification hash function;

storing said file in said database at a location identified by said random number; and

adding said random number and said verification hash value to an entry in a data structure associated with said database, wherein said random number and said verification hash value are associated with one another in said data structure, whereby the addition of said entry in said data structure indicates the presence of said file in said database.

20. A method as recited in claim 19 further comprising:

determining whether a copy of said file already exists in said database by searching said data structure for a value that matches said verification hash value.

21. A method as recited in claim 19 wherein said data structure is a table, a list or a tree data structure.

22. A method of adding a computer file to a database of files using a random number generator, said method comprising:

receiving said computer file to be added to said database;

accessing a data structure representing said database, said data structure including a random number-verification hash value pair for each of said files in said database, wherein said random numbers have been computed using said random number generator, and wherein said verification hash values have been computed using a verification hash function;

computing a new verification hash value for said file using said verification hash function;

determining whether a copy of said file already exists in said database by searching said data structure for a value that matches said new verification hash value;

when it is determined that a copy of said file already exists in said database, returning a random number associated with said copy of said file to a user, said random number being associated with said new verification hash value.

23. A method as recited in claim 22 wherein said data structure is a table, a list or a tree data structure.

24. A method of adding a computer file to a database of files using a random number generator, said method comprising:

receiving said computer file to be added to said database;

generating a first random number for said file using said random number generator;

accessing a data structure representing said database, said data structure including a random number-verification hash value pair for each of said files in said database, wherein said random numbers have been computed using said random number generator, and wherein said verification hash values have been computed using a verification hash function;

computing a new verification hash value for said file using said verification hash function;

determining whether a copy of said file already exists in said database by searching said data structure for a value that matches said new verification hash value; and

when it is determined that a copy of said file already exists in said database,

adding said first random number and said new verification hash value as an entry in said data structure,

retrieving a second random number from said data structure that is associated with said copy of said file,

adding a mapping to said entry that maps said first random number to said second random number, whereby a reference to said first random number is mapped to said second random number to facilitate access to said file.

25. A method as recited in claim 24 wherein said data structure is a table, a list or a tree data structure.

26. A method of retrieving a desired computer file from a database using a random number generator, said method comprising:

obtaining a random number associated with said desired computer file, said random number having been generated for said file using said random number generator;

retrieving a stored file from said database using said random number as a reference;

using said random number to lookup an associated first hash value from a data structure associated with said database, wherein said first hash value has been computed from said stored file using a verification hash function;

computing a second hash value for said stored file using said verification hash function;

comparing said first hash value to said second hash value; and

determining that said stored file is said desired file when said first hash value matches said second hash value, whereby said desired file is retrieved from said database.

27. A method as recited in claim 26 further comprising:

when it is determined that said stored file is said desired file, delivering said stored file to a user.

28. A method as recited in claim 26 further comprising:

performing said step of retrieving a stored file by searching said data structure for said random number.

29. A method as recited in claim 26 wherein said data structure is a table, a list or a tree data structure.